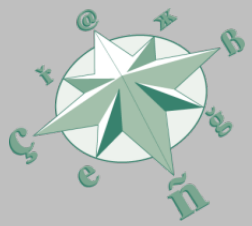
UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

ALeF: Active Learning Framework for Readability Prediction

Philip van Oosten — Véronique Hoste

LT^3, Language and Translation Technology Team

Atila, September 23, 2010



Overview

1. Problem setting
   - In general
   - Our proposed solution for readability prediction
2. The Wolfit workflow engine
   - Wolfit API
   - Wolfit Editor screenshots
3. ALeF: active learning on top of Wolfit
   - Cascaded approach
   - With active learning: ALeF
   - Solutions provided by ALeF

UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Overview

Overhead for NLP experiments

Infrastructure

- setting up software infrastructure: in many cases overhead or done manually
- maintenance of complex tasks $\rightarrow$ overhead.

Resources

- corpus annotation often necessary
- data sets often used in isolation $\Longrightarrow$ experimental results often incomparable
- data set used in more tasks or settings $\Longrightarrow$ more valuable to compare methodologies
- more data sets per task $\Longrightarrow$ better comparison of methodologies

UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Possible solutions

Infrastructure

- applications of which small parts can be adapted.
- workflow engines
- ...

Resources

- share resources
- use same data in multiple settings
- ...

UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Our solution for Readability Prediction

Readability prediction is a small domain

$\Longrightarrow$ try to provide solutions that can be generalized to other domains
(but first focus on readability)

UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

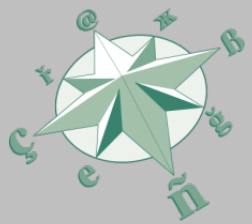
Our solution for Readability Prediction

Readability prediction is a small domain

$\Longrightarrow$ try to provide solutions that can be generalized to other domains
(but first focus on readability)

- a workflow engine built to reduce overhead
- built on top of that: ALeF, a complete framework for readability research

Status: implementation of the workflow engine



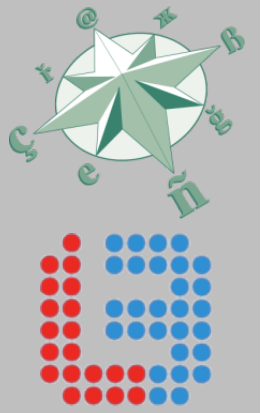
UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Overview

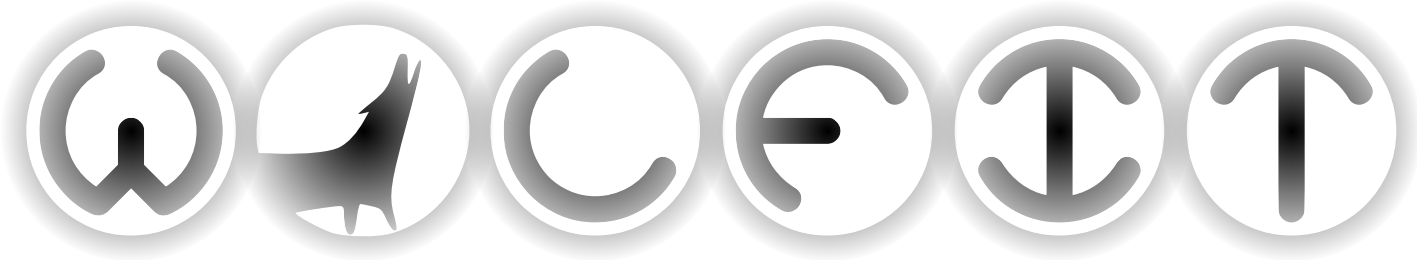
Advantages of workflow engines?

Separation of concerns, so that users need only minimal knowledge of

- multiple programming languages
- pipelining/parallellization
- in-detail usage of resources



UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION



A workflow engine focused on rapid development

- API to construct and execute workflows
- web application to construct workflows (django-wolfit)
- GNU GPLv2 on Google Code

`http://code.google.com/p/wolfit`
`http://code.google.com/p/django-wolfit`

UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Wolfit API

Functionality:

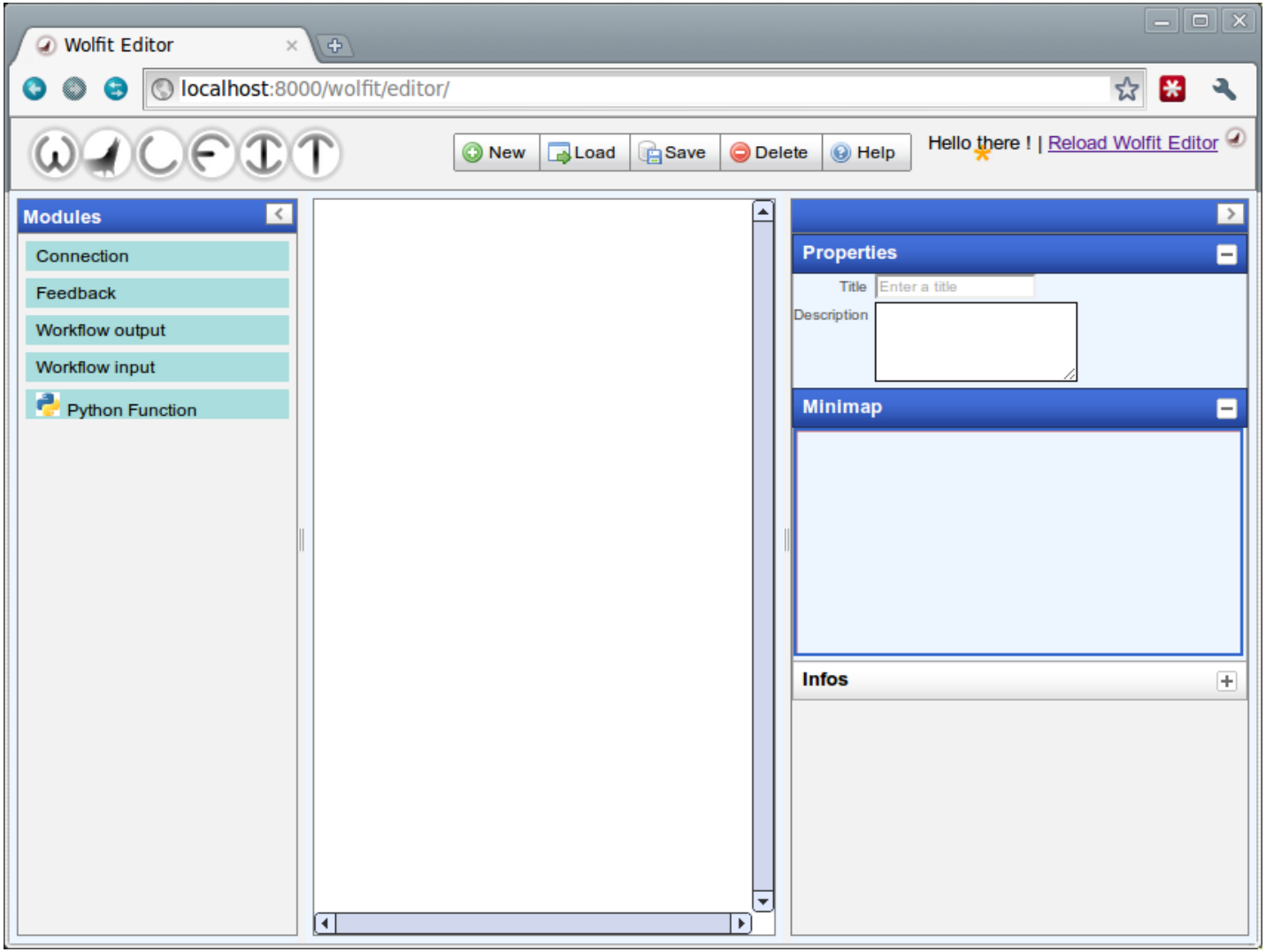
- construct/configure/execute workflows
- expose workflows as Python functions
- import and export
- visualization

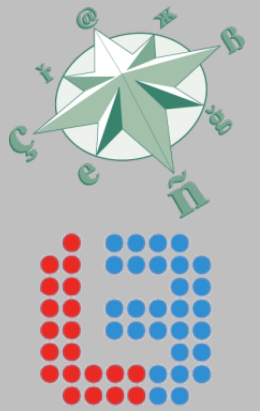
To do:

- abstract workflows
- more task types: CLI, CObject, Java, ...
- workflows as tasks in other workflows
- more control structures
- ...

UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

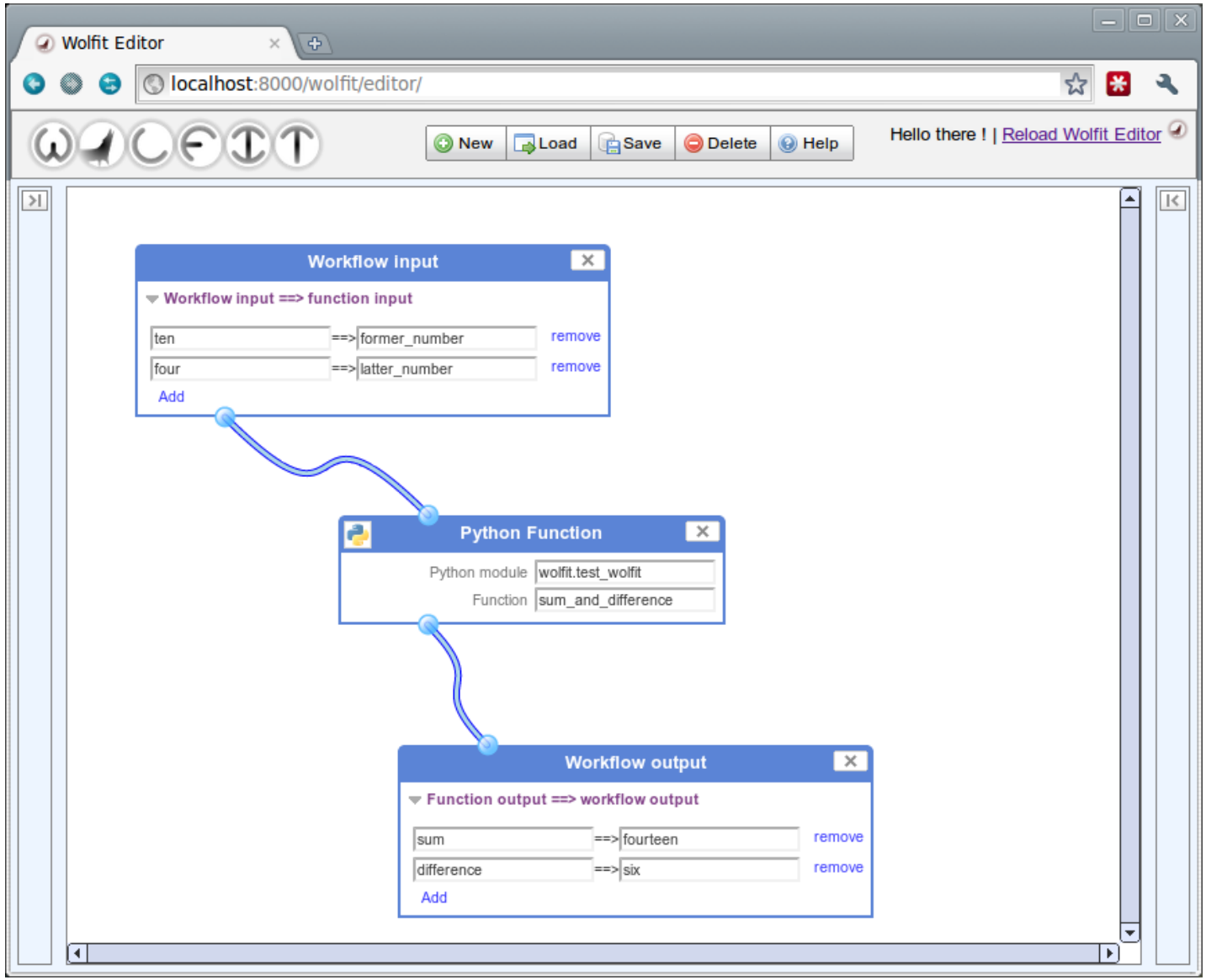
An empty editor. Left: available modules
Right: Name, description, minimap, info
Top: New, load, save, delete, . . . buttons





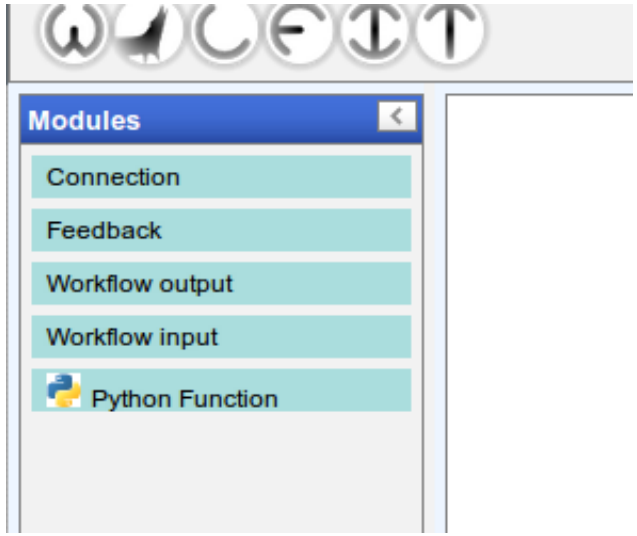
UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

A function connected to workflow input and output.



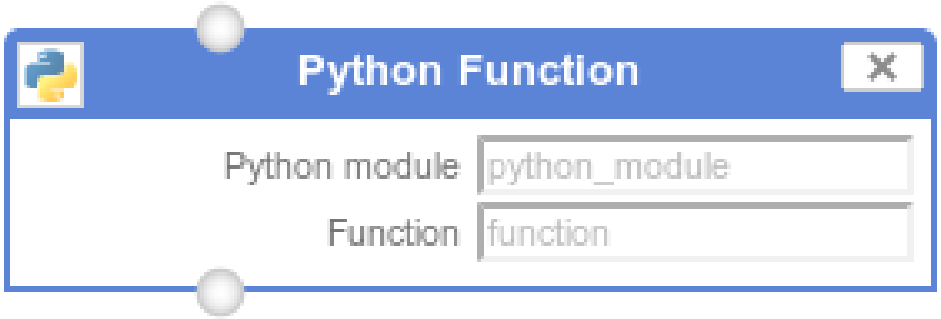
UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

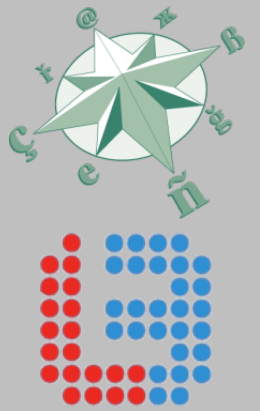
The currently available modules.



UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

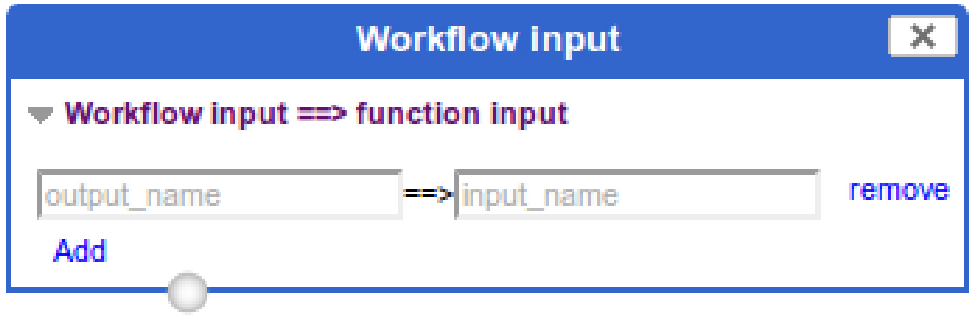
A python function, taking keyword arguments as input and a dictionary as return value.

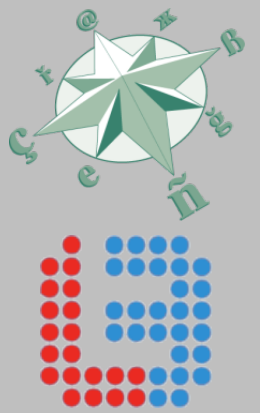




UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

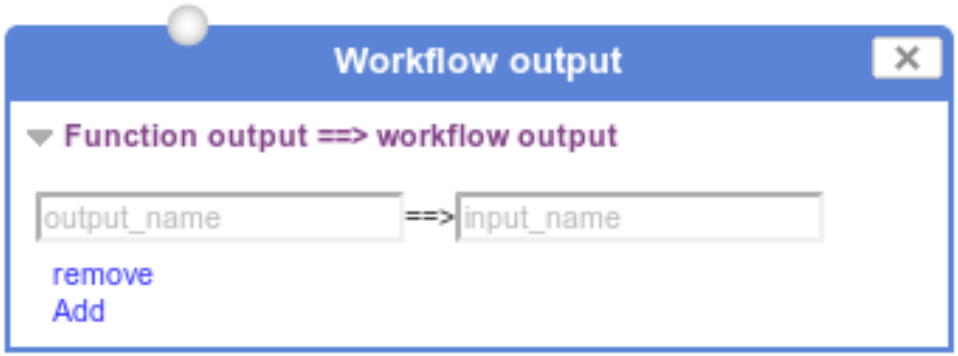
A workflow input





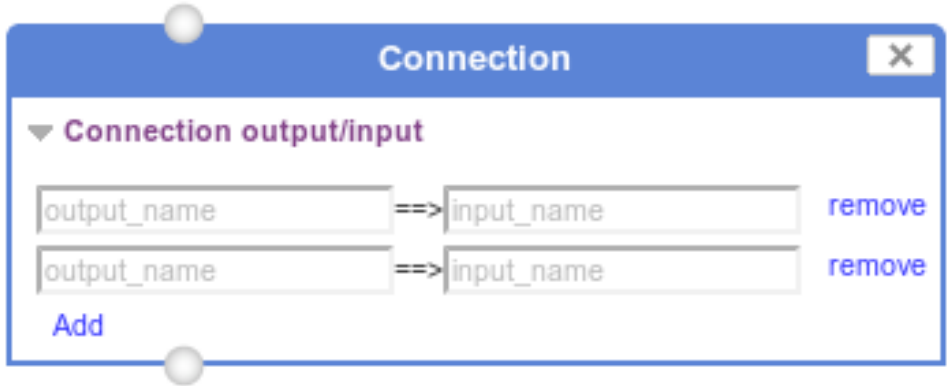
UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

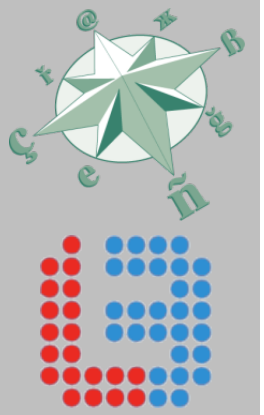
A workflow output



UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

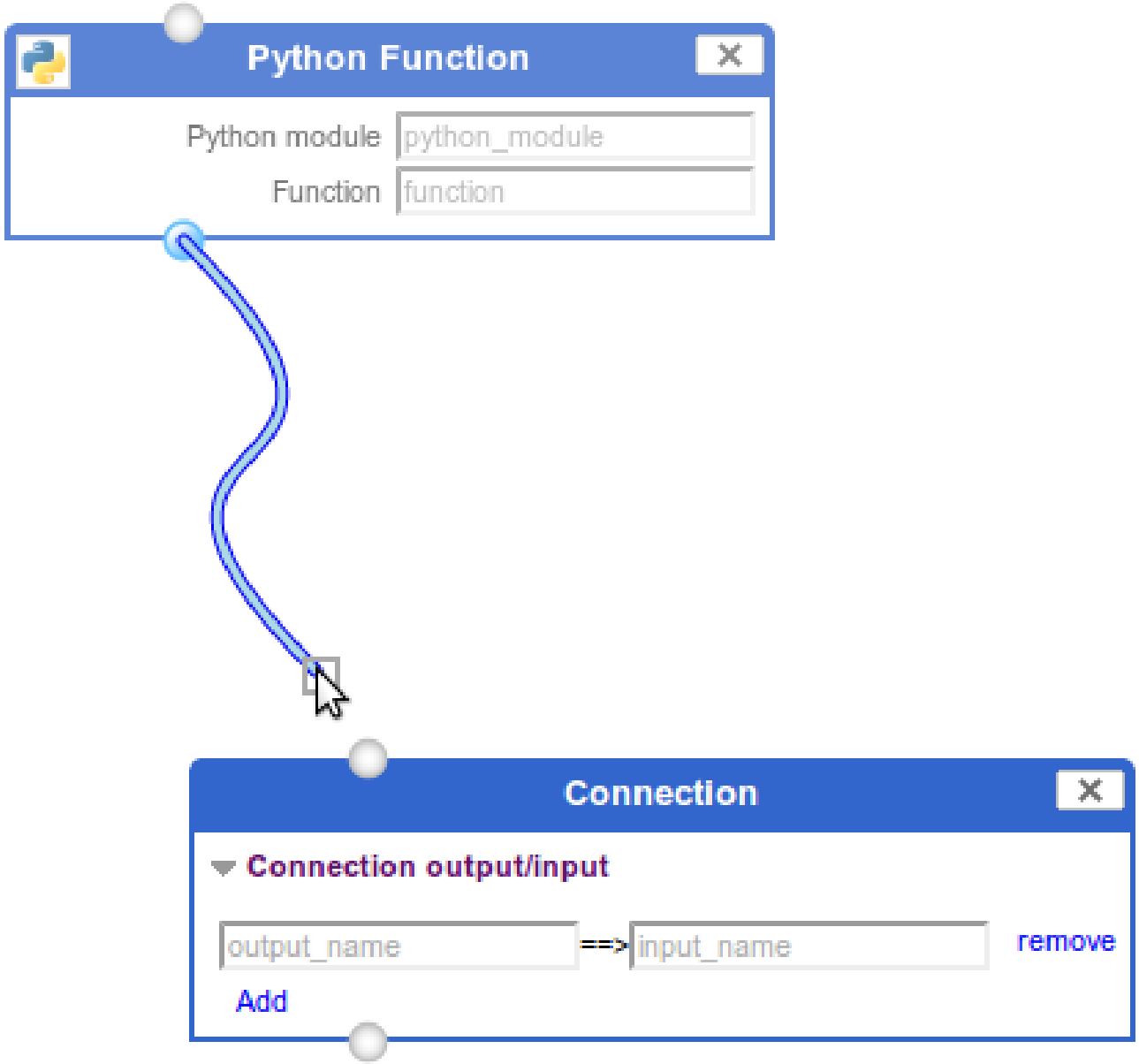
A connection between two functions

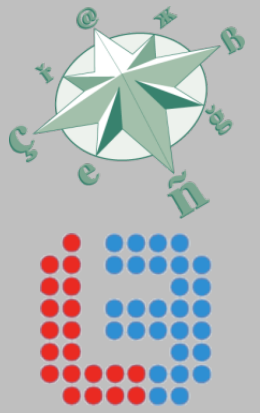




UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

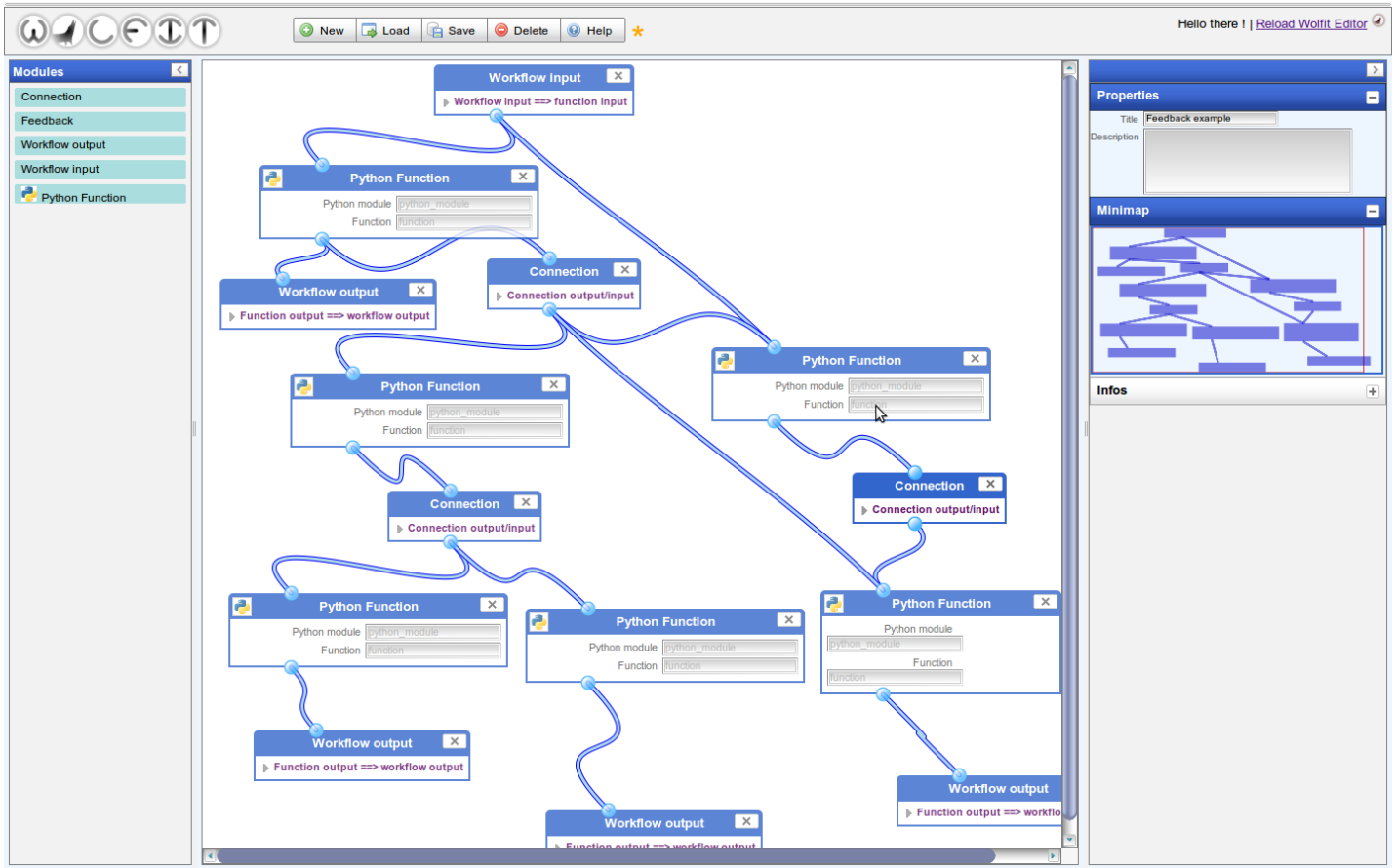
You can connect modules by dragging from one *terminal* to another.

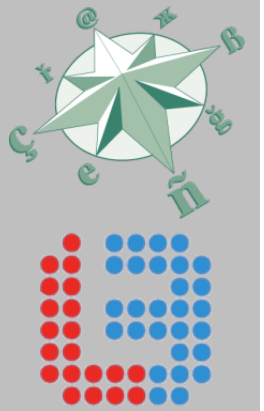




UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

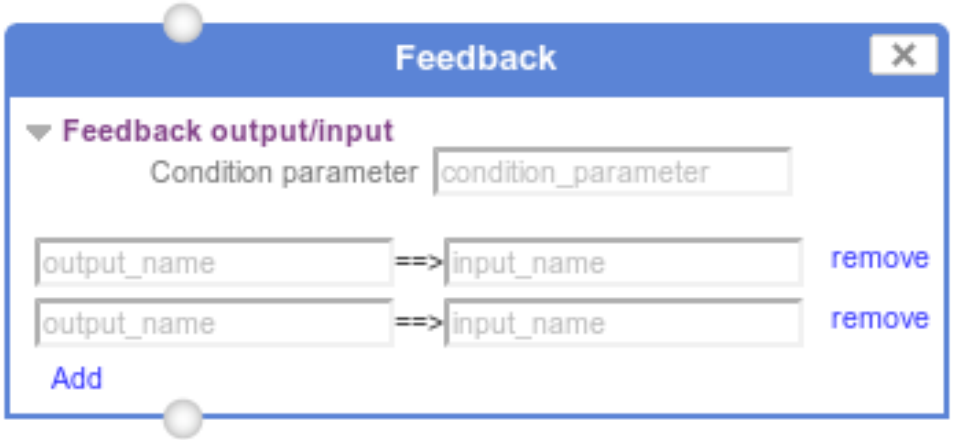
An example of what a more elaborate workflow can look like.

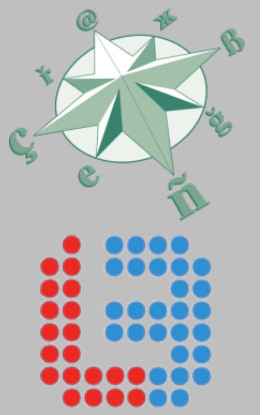




UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

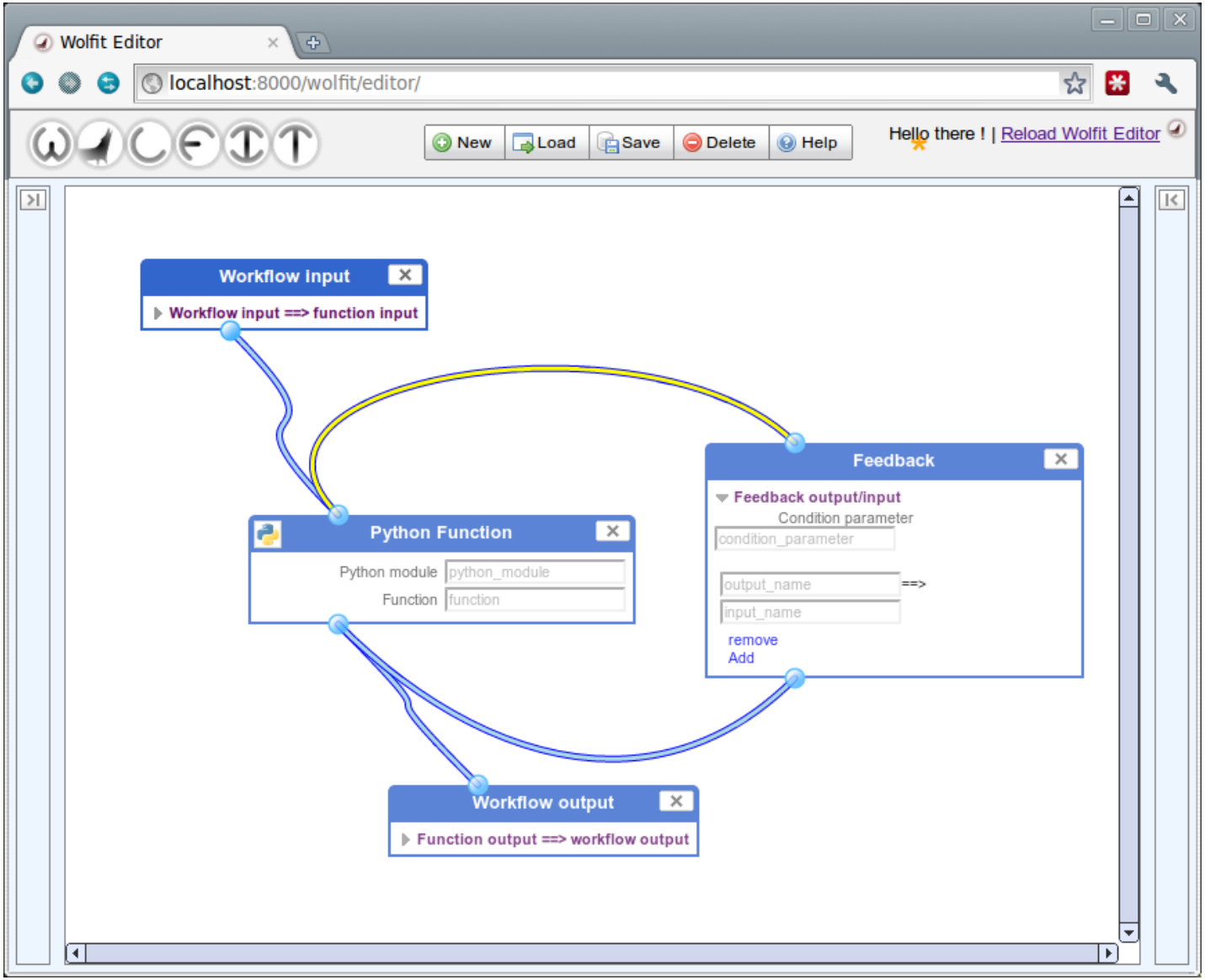
A feedback module





UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Feedback loops are like while loops. They enable active learning, backtracking, heuristic algorithms, . . .



UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Creating and executing a workflow.

By means of a Python script:

- Use the API directly to compose a Process object
- Configure a runner. Currently the ThreadPoolRunner is preferrable.
- Execute the workflow like you would execute a Python function

By means of django-wolfit:

- Create a workflow using the Django web application
- Use an *importer* to create a Process object.
- Configure a runner.
- Execute the Process like a python function.

To do: make creating web demos based on workflows easy

UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Overview

Cascaded approach

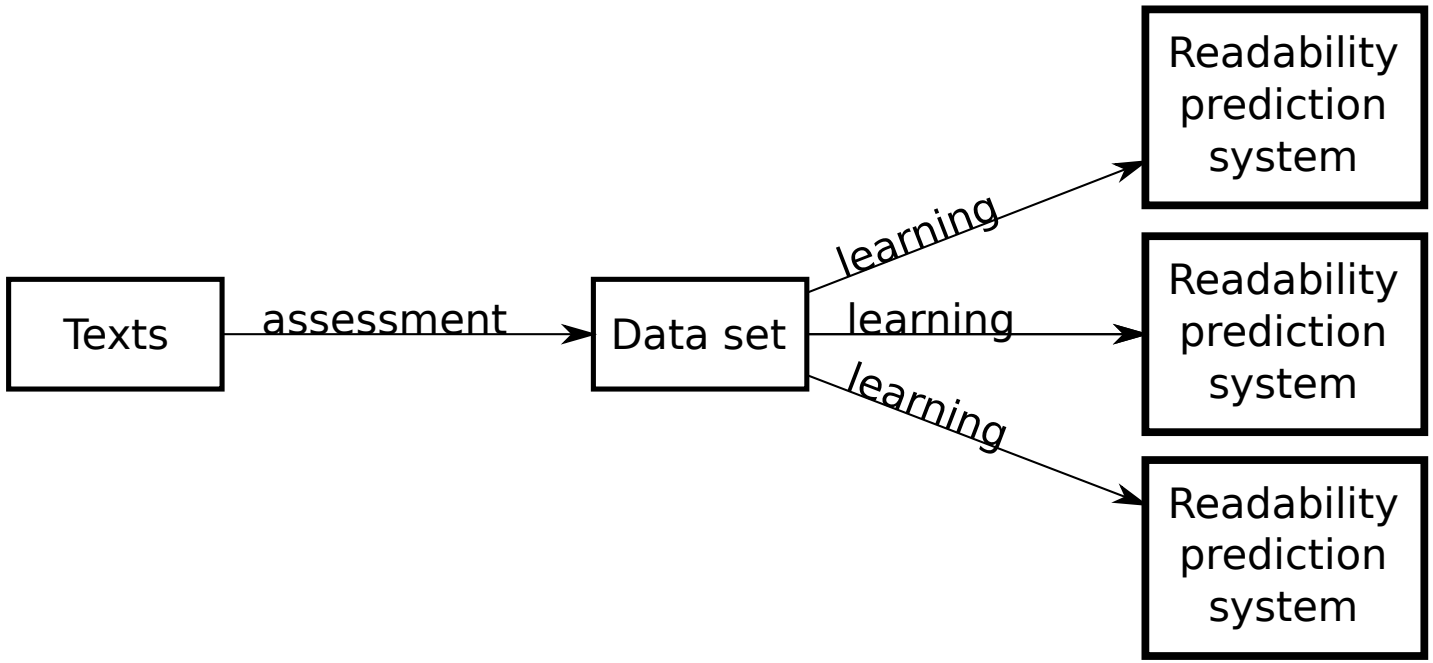
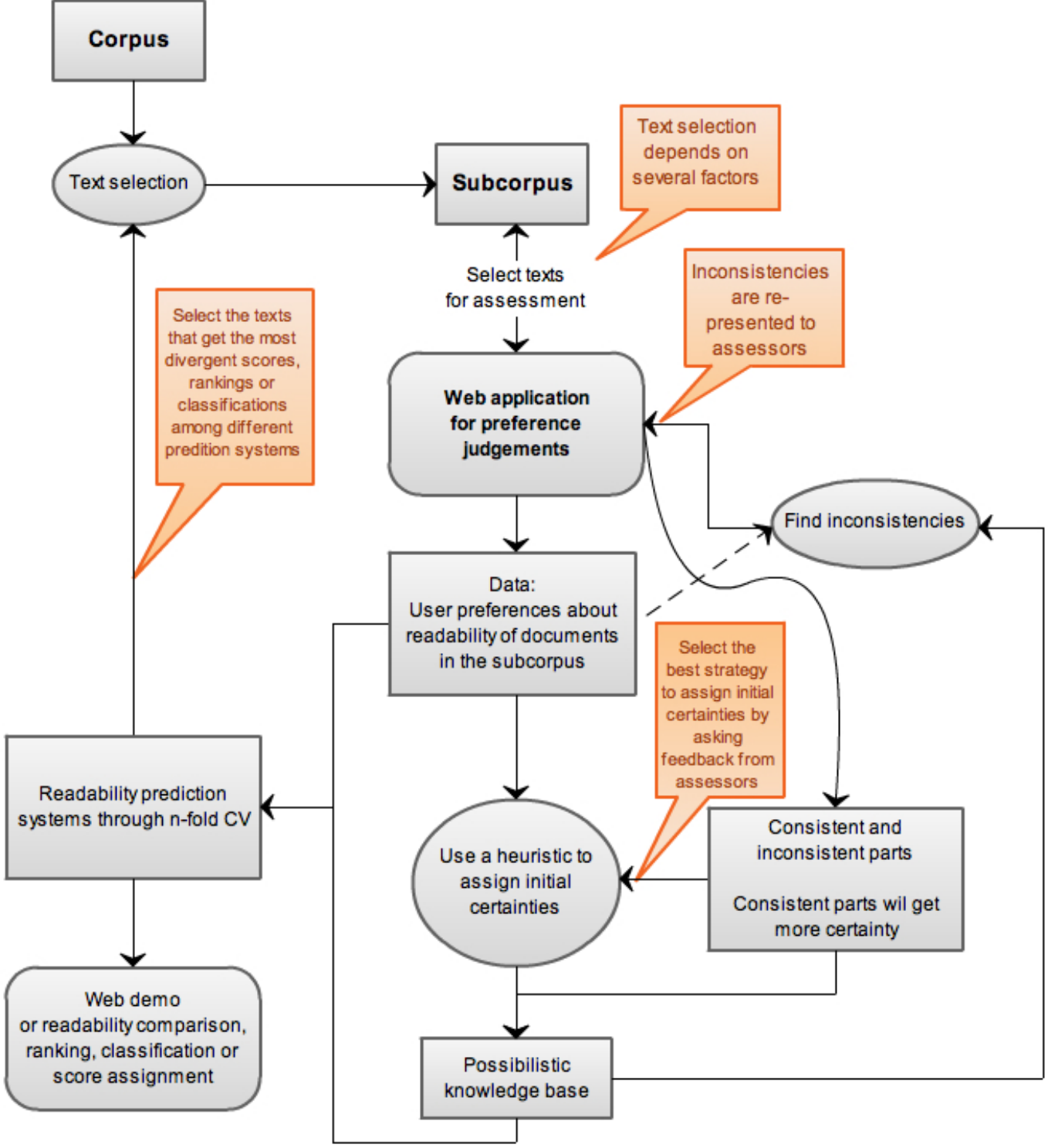


Figure: Readability prediction as a typical cascaded CL task.

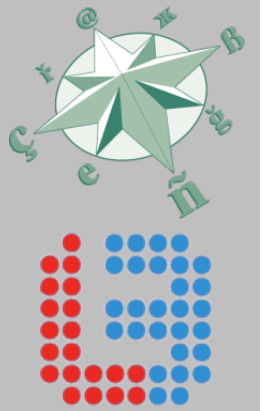


UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Solutions provided by ALeF

Infrastructure An extensible application was presented and will be implemented. Third party plugins can be added.

Resources Encourage people to make use of ALeF and donate their data sets for research (first try the framework ourselves).



UNIVERSITY COLLEGE GHENT
MEMBER OF GHENT UNIVERSITY ASSOCIATION

Thank you!